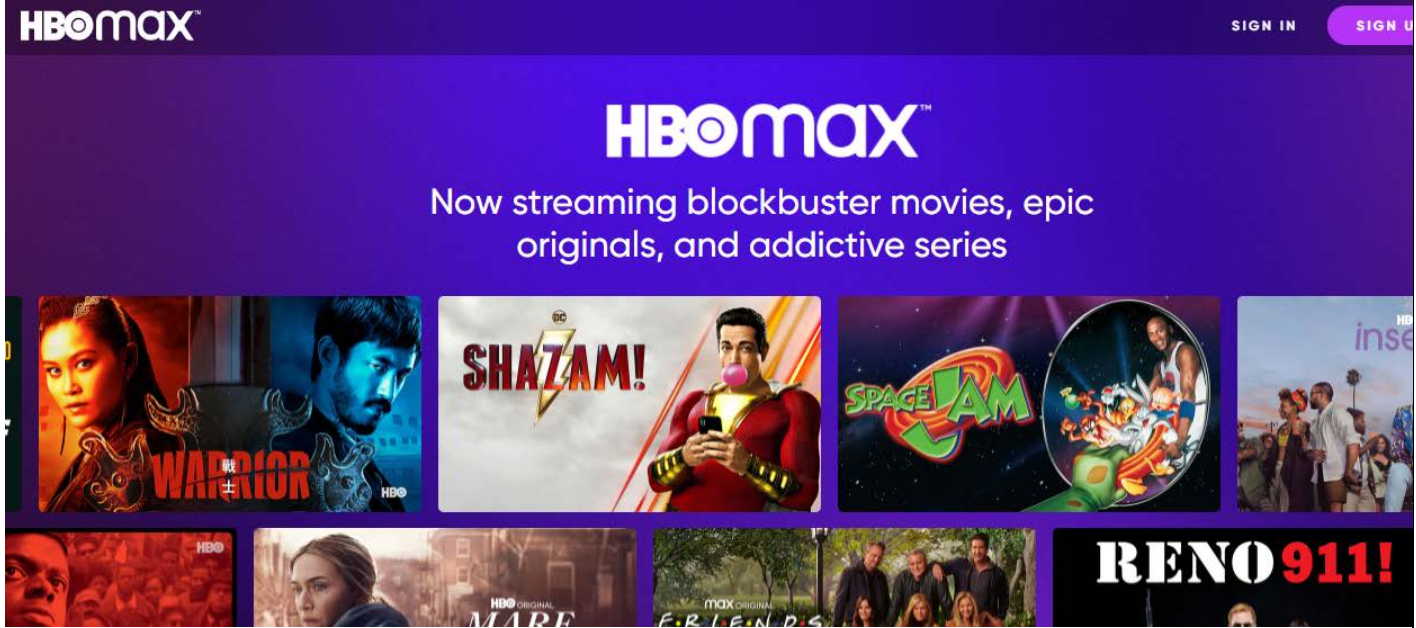


EXHIBIT D

US8195828B2	HBO
<p>1. A method for discontinuous transmission, in sections, of encoded video data in a network of distributed appliances, the method comprising the following steps:</p>	<p>HBO utilizes the MPEG-DASH streaming protocol (“the Standard”), including for delivery of VOD contents to its viewers/customers. As shown below, a video content from HBO is streamed and the data traffic is captured showing the media format as MPEG-DASH, and the MPD file, (e.g., the Media Presentation Description file is used by MPEG-DASH to contain information about the media playing). In addition, HBO provides trick mode operation (such as 15 Sec forward/reverse, etc.) to the streamed video.</p> <p>The MPEG-DASH standard practices a method for discontinuous transmission (e.g., discontinues transmission of video data during trick mode in which a reduced frame rate representation or an I-frame only version is transmitted), in sections (e.g., media segments), of encoded video data (e.g., Codec encoded representations of video stream) in a network of distributed appliances (e.g., network of DASH clients and servers).</p>  <p>https://www.hbomax.com/</p>

Shown below is the URL of an .mpd master file sent by HBO which identifies the usage of MPEG-DASH based streaming by HBO. The .mpd master file refers to all the variants of the video encoded for various bandwidths and resolutions. The URL of .mpd master file is:
https://cmf.l3.us.hbomaxcdn.com/videos/GYNNHNwDfl6yYnQEAAAZ/0/00077c/0_JvEEZg_noanc_aBToBqQ_1080hi.mpd

The screenshot displays the Progress Telerik Fiddler Web Debugger interface. The left pane shows a list of network sessions, with session 111 selected. The main pane shows the details of this session, including the URL and the JSON response body. The JSON body contains a 'cdn' object with 'provider=LEVEL3' and 'containerType=DASH'. The 'manifest' field is highlighted with a red box, showing the URL: https://cmf.l3.us.hbomaxcdn.com/videos/GYNNHNwDfl6yYnQEAAAZ/0/00077c/0_JvEEZg_noanc_aBToBqQ_1080hi.mpd. The 'manifestType' is also highlighted with a red box, showing 'manifestType=SingleVideo'.

#	Result	Protocol	Host	URL
82	200	HTTP	Tunnel to artist.api.cdn.hbo.com:443	
83	200	HTTP	Tunnel to artist.api.cdn.hbo.com:443	
84	200	HTTP	Tunnel to artist.api.cdn.hbo.com:443	
85	200	HTTP	Tunnel to artist.api.cdn.hbo.com:443	
86	302	HTTPS	9848493.fls.double...	/activity;src=9848493;ty...
87	302	HTTPS	9848493.fls.double...	/activity;src=9848493;ty...
88	302	HTTPS	9848493.fls.double...	/activity;src=9848493;ty...
89	200	HTTPS	9848493.fls.double...	/activity;dc_pre=CMuelt...
90	200	HTTPS	comet.api.hbo.com	/content
91	200	HTTPS	9848493.fls.double...	/activity;dc_pre=CISant...
92	200	HTTPS	9848493.fls.double...	/activity;dc_pre=CP_Dtd...
93	207	HTTPS	comet.api.hbo.com	/content
94	200	HTTP	adservice.google.com:443	
95	200	HTTP	Tunnel to comet.api.hbo.com:443	
96	200	HTTP	Tunnel to telegraph.api.hbo.com:443	
97	200	HTTP	Tunnel to telegraph.api.hbo.com:443	
98	200	HTTPS	comet.api.hbo.com	/express-content/urn:hbo...
99	200	HTTP	adservice.google.com:443	
100	200	HTTPS	adservice.google.com	/ddm/fls/z/dc_pre=CMuelt...
101	200	HTTPS	comet.api.hbo.com	/express-content/urn:hbo...
102	200	HTTPS	comet.api.hbo.com	/express-content/urn:hbo...
103	201	HTTPS	telegraph.api.hbo.c...	/events/v1
104	200	HTTPS	adservice.google.com	/ddm/fls/z/dc_pre=CISant...
105	200	HTTPS	comet.api.hbo.com	/express-content/urn:hbo...
106	201	HTTPS	telegraph.api.hbo.c...	/events/v1
107	200	HTTP	Tunnel to www.googleadservices.co...	
108	200	HTTPS	adservice.google.com	/ddm/fls/z/dc_pre=CP_Dt...
109	201	HTTPS	telegraph.api.hbo.c...	/events/v1
110	201	HTTPS	telegraph.api.hbo.c...	/events/v1
111	207	HTTPS	comet.api.hbo.com	/content
112	200	HTTPS	artist.api.cdn.hbo.c...	/images/oJVB1Dy4wqeVo...

Source: Packet captures by Fiddler

Request Headers [Raw] [Header Definitions]

GET /videos/GX10dgepEp4_wwEAABpp/0/f3c116/0_KRsEnw_noanc_aBToBqQ_1080hi.mpd HTTP/1.1

Client

Accept: */*
 Accept-Encoding: gzip, deflate, br
 Accept-Language: en-US,en;q=0.9
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36

Miscellaneous

Referer: https://play.hbomax.com/

Security

Origin: https://play.hbomax.com
 sec-ch-ua: "Not;A Brand";v="99", "Google Chrome";v="91", "Chromium";v="91"
 sec-ch-ua-mobile: ?0

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies Raw

JSON XML

MPD [mediaPresentationDuration=PT44.04399871826172S xmlns=urn:mpeg:dash:schema:mpd:2011 minBufferTime=PT2S type=static profile

Period [id=0 start=PT0.000000S]

AdaptationSet [id=0 lang=en-US contentType=audio subsegmentAlignment=true subsegmentStartsWithSAP=1]

Representation [id=a2 bandwidth=384216 mimeType=audio/mp4 codecs=ac-3 audioSamplingRate=48000]

BaseURL

a/a2.mp4

<AudioChannelConfiguration value="2" schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" xmlns="urn:mpeg:dash:schema:mpd:2011" />

SegmentBase [xmlns=urn:mpeg:dash:schema:mpd:2011 indexRange=576-751 timescale=48000]

<Initialization xmlns="urn:mpeg:dash:schema:mpd:2011" range="0-575"></Initialization>

AdaptationSet [id=1 lang=en-US contentType=audio subsegmentAlignment=true subsegmentStartsWithSAP=1]

Representation [id=a0 bandwidth=256216 mimeType=audio/mp4 codecs=ec-3 audioSamplingRate=48000]

BaseURL

a/a0.mp4

<AudioChannelConfiguration value="2" schemeIdUri="urn:mpeg:mpegB:ac3:ChannelConfiguration" xmlns="urn:mpeg:dash:schema:mpd:2011" />

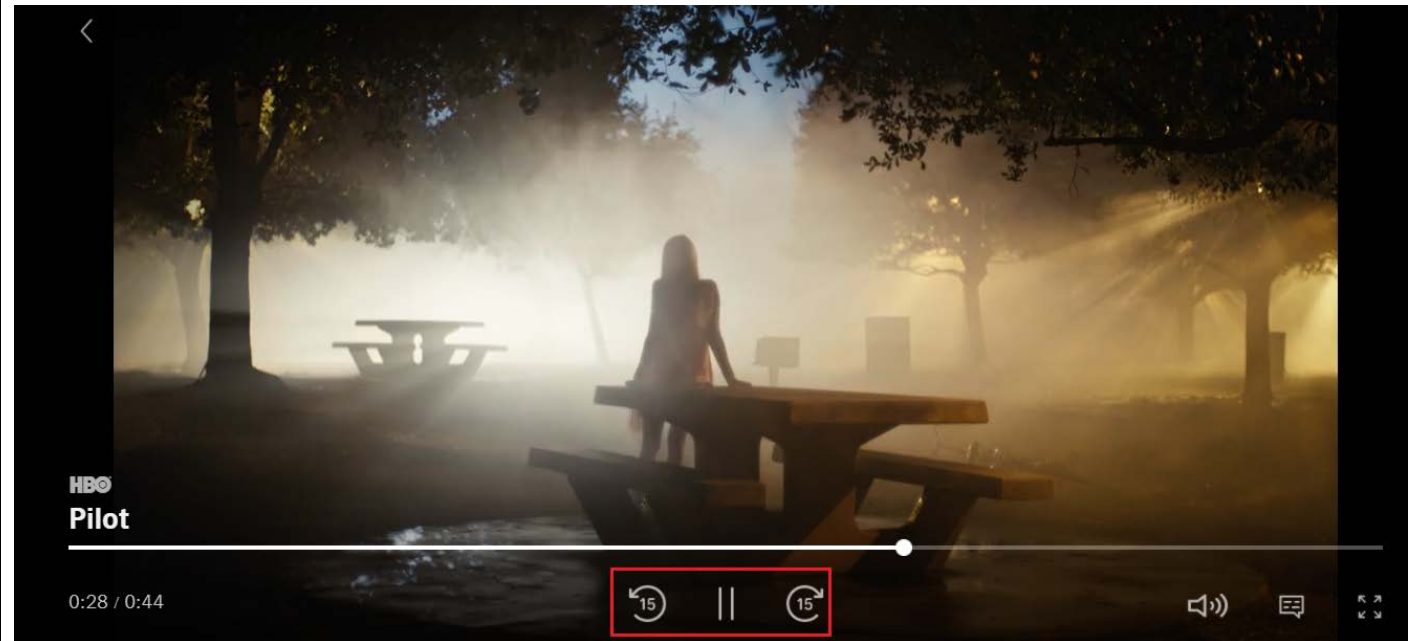
SegmentBase [xmlns=urn:mpeg:dash:schema:mpd:2011 indexRange=578-753 timescale=48000]

<Initialization xmlns="urn:mpeg:dash:schema:mpd:2011" range="0-577"></Initialization>

Expand All Collapse

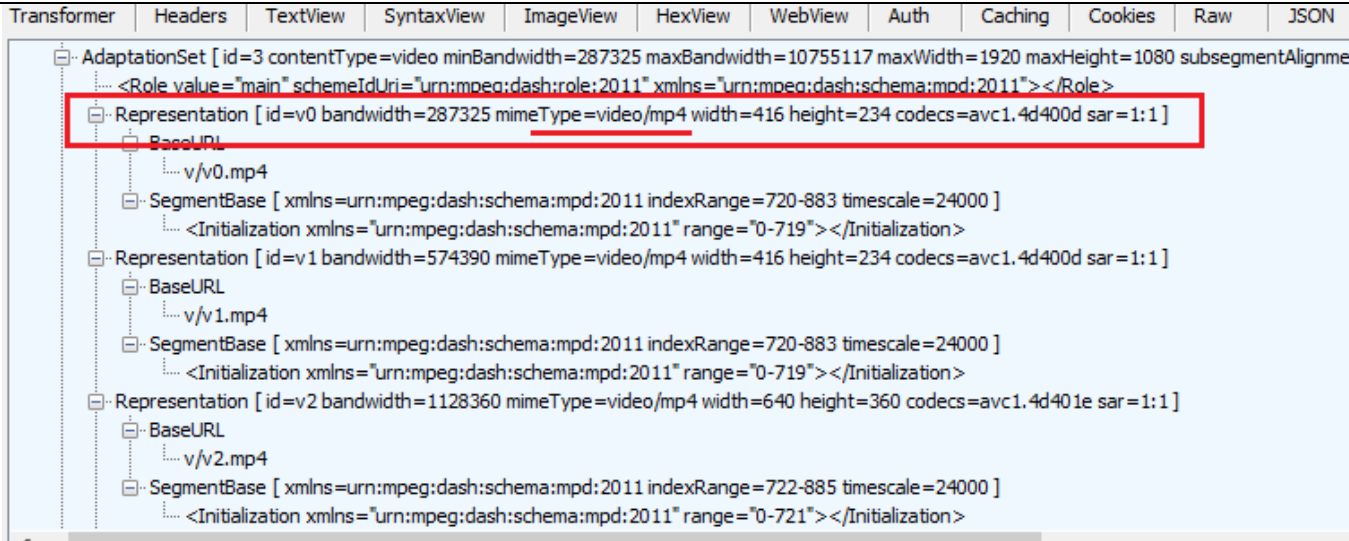
Source: Packet captures by Fiddler

HBO, as depicted below, provides trick mode operation (such as 15 Sec forward/reverse, etc.) to the streamed video.



https://play.hbomax.com/page/urn:hbo:page:GXKN_xQX5csPDwwEAAABj:type:series

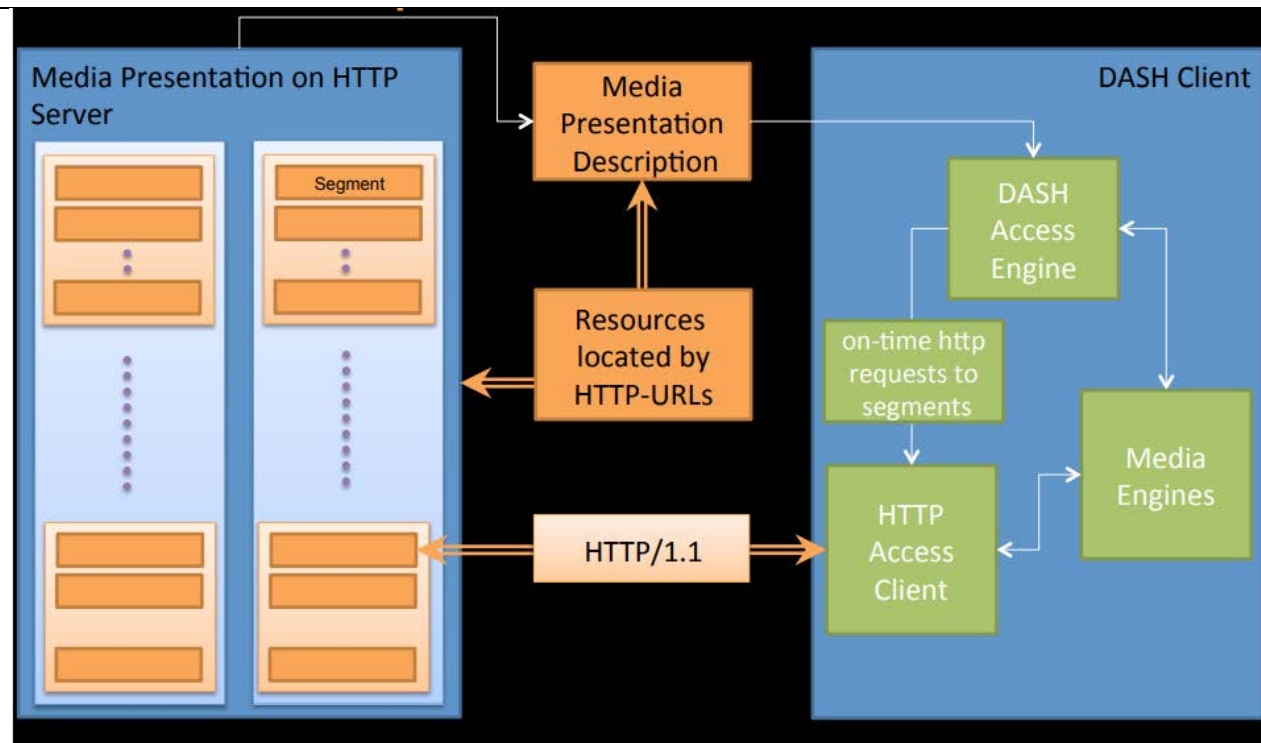
The HBO server streams video using AVC1/H.264 codec (e.g., video encoding) as shown below.



The screenshot displays the JSON structure of an AdaptationSet from a packet capture. The top tabs include Transformer, Headers, TextView, SyntaxView, ImageView, HexView, WebView, Auth, Caching, Cookies, Raw, and JSON. The JSON tree shows an AdaptationSet with id=3 and contentType=video. It contains three Representations: id=v0, id=v1, and id=v2. The first Representation (id=v0) is highlighted with a red box. Below it, the SegmentBase and Initialization sections are visible.

```
AdaptationSet [ id=3 contentType=video minBandwidth=287325 maxBandwidth=10755117 maxWidth=1920 maxHeight=1080 subsegmentAlignmer
  <Role value="main" schemeIdUri="urn:mpeg:dash:role:2011" xmlns="urn:mpeg:dash:schema:mpd:2011"></Role>
  Representation [ id=v0 bandwidth=287325 mimeType=video/mp4 width=416 height=234 codecs=avc1.4d400d sar=1:1 ]
    BaseURL
      v/v0.mp4
    SegmentBase [ xmlns=urn:mpeg:dash:schema:mpd:2011 indexRange=720-883 timescale=24000 ]
      <Initialization xmlns="urn:mpeg:dash:schema:mpd:2011" range="0-719"></Initialization>
    Representation [ id=v1 bandwidth=574390 mimeType=video/mp4 width=416 height=234 codecs=avc1.4d400d sar=1:1 ]
      BaseURL
        v/v1.mp4
      SegmentBase [ xmlns=urn:mpeg:dash:schema:mpd:2011 indexRange=720-883 timescale=24000 ]
        <Initialization xmlns="urn:mpeg:dash:schema:mpd:2011" range="0-719"></Initialization>
    Representation [ id=v2 bandwidth=1128360 mimeType=video/mp4 width=640 height=360 codecs=avc1.4d401e sar=1:1 ]
      BaseURL
        v/v2.mp4
      SegmentBase [ xmlns=urn:mpeg:dash:schema:mpd:2011 indexRange=722-885 timescale=24000 ]
        <Initialization xmlns="urn:mpeg:dash:schema:mpd:2011" range="0-721"></Initialization>
```

Source: Packet captures by Fiddler



<https://www.w3.org/2011/09/webtv/slides/W3C-Workshop.pdf>

3.2.9. Trick Mode Support

Trick Modes are used by DASH clients in order to support fast forward, seek, rewind and other operations in which typically the media, especially video, is displayed in a speed other than the normal playout speed. In order to support such operations, it is recommended that the content author adds Representations at lower frame rates in order to support faster playout with the same decoding and rendering capabilities.

However, Representations targeted for trick modes are typically not be suitable for regular playout. If the content author wants to explicitly signal that a Representation is only suitable for trick mode cases, but not for regular playout, the following is recommended:

- add one or multiple Adaptation Sets that that only contains trick modes Representations
- annotate each Adaptation Set with an **EssentialProperty** descriptor or **SupplementalProperty** descriptor with URL "<http://dashif.org/guidelines/trickmode>" and the @value the value of @id attribute of the Adaptation Set to which these trick mode Representations belong. The trick mode Representations must be time-aligned with the Representations in the main Adaptation Set. The value may also be a white-space separated list of @id values. In this case the trick mode Adaptation Set is associated to all Adaptation Sets with the values of the @id.
- signal the playout capabilities with the attribute @maxPlayoutRate for each Representation in order to indicate the accelerated playout that is enabled by the signaled codec profile and level.
- If the Representation is encoded without any coding dependency on the elementary stream level, i.e. each sample is a SAP type 1, then it is recommended to set the @codingDependency attribute to FALSE.
- If multiple trick mode Adaptation Sets are present for one main Adaptation Set, then sufficient signaling should be provided to differentiate the different trick mode Adaptation Sets. Different Adaptation Sets for example may be provided as thumbnails (low spatial resolution), for fast forward or rewind (no coding dependency with @codingDependency set to false and/or lower frame rates), longer values for @duration to improve download frequencies or different @maxPlayoutRate values. Note also that the @bandwidth value should be carefully documented to support faster than real-time download of Segments.

	<h2>4.10. Trick Mode for Live Services</h2> <h3>4.10.1. Introduction</h3> <p>In certain use cases, along with the offering of the main content, a content author also wants to provide <u>a trick mode version primarily of the video Adaptation Set along with the live content that can be used for rewind and fast forward in the time shift buffer of the Media Presentation.</u> In section 3.2.9 signalling is introduced to flag and customize Adaptation Sets for Trick Modes. This clause provides additional service offering requirements and recommendations for trick modes in case of a live service. <u>Typically, a reduced frame rate Representation or an I-frame only version is provided for supporting such trick mode operations.</u></p> <p>https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf</p> <p>In On Demand Profile Periods, Sub-Representations in combination with Index Segments and Subsegment Index boxes may be used for efficient trick mode implementation. <u>Sub-Representations in combination with Index Segments containing Segment Index boxes ('sidx') and Subsegment Index boxes ('ssix') may be used for trick mode implementation. Given a Sub-Representation with the desired @maxPlayoutRate, ranges corresponding to SubRepresentation@level all level values from SubRepresentation@dependencyLevel may be extracted via byte ranges constructed from the information in Subsegment Index Box. These ranges can be used to construct more compact HTTP GET request.</u></p> <p>https://www.etsi.org/deliver/etsi_ts/103200_103299/103285/01.02.01_60/ts_103285v010201p.pdf</p>
<p>creation of an HTTP GET request for requesting a fast search operation of an original video stream, the request stating a playback speed parameter and an initial position and optionally at least one parameter selected from a group of parameters consisting</p>	<p>The Standard practices creation of an HTTP GET request (e.g., an HTTP request by the DASH client) for requesting a fast search operation (e.g., trick mode operation such as fast forward or rewind) of an original video stream (e.g., a video stream), the request stating a playback speed parameter (e.g., playout rate) and an initial position (e.g., position of a segment such as its index number, time, or byte range when the trick mode is requested) and optionally at least one parameter selected from a group of parameters consisting of file name, file type, path (e.g., a URL of requested segments for trick mode operation), and playback direction.</p> <p>During trick mode request operation, the DASH clients select from MPD the Adaptations/Representations/Segments/Subsegments related to trick mode based on requested playback</p>

of file name, file type, path, and playback direction;

speed parameter (e.g., @MaxPlayout rate) and the position (e.g., index, time, or byte range) of the media segment when the trick mode is requested, and downloads the selected trick-mode-related segments by HTTP GET request by using the assigned URL of the requested segments.

DASH Client

A DASH client retrieves and plays the video content using these steps: howstuffworks.com

1. The client downloads and reads the **Media Presentation Description (MPD)** to get important information, such as the content locations, segment encodings, resolution, minimum and maximum bandwidths, accessibility features like closed captioning and content restrictions (such as DRM).
2. The client selects an appropriate segment encoding and begins streaming the content through a series of HTTP requests. The server creates and encodes each segments on demand for each request, all from the same source.
3. The client buffers data as it's downloaded, while also keeping track of fluctuations in the connection bandwidth. If necessary, the client automatically changes to a different segment encoding (from those listed by the MPD) that's more compatible with the current bitrate. This ensures the client maintains a sufficient buffer throughout the video without downloading more data than you need.

https://www.bogotobogo.com/VideoStreaming/mpeg_dash.php

Figure 2 shows the DASH aspects of a DASH-AVC/264 client compared to a client supporting all features of the DASH ISO BMFF Live and On-Demand profile. The main supported features are:

- support of HTTP GET and partial GET requests to download Segments and Subsegments
- three different addressing schemes: number and time-based templating as well as byte range based requests.
- support of metadata as provided in the MPD and Segment Index
- download of Media Segments, Initialization Segments and Segment Index
- ISO BMFF parsing
- synchronized presentation of media components from different Adaptation Sets
- switching of video streams at closed GOP boundaries

3.2.9. Trick Mode Support

Trick Modes are used by DASH clients in order to support fast forward, seek, rewind and other operations in which typically the media, especially video, is displayed in a speed other than the normal playout speed. In order to support such operations, it is recommended that the content author adds Representations at lower frame rates in order to support faster playout with the same decoding and rendering capabilities.

However, Representations targeted for trick modes are typically not be suitable for regular playout. If the content author wants to explicitly signal that a Representation is only suitable for trick mode cases, but not for regular playout, the following is recommended:

- add one or multiple Adaptation Sets that only contains trick modes Representations
- annotate each Adaptation Set with an **EssentialProperty** descriptor or **SupplementalProperty** descriptor with URL "<http://dashif.org/guidelines/trickmode>" and the @value the value of @id attribute of the Adaptation Set to which these trick mode Representations belong. The trick mode Representations must be time-aligned with the Representations in the main Adaptation Set. The value may also be a white-space separated list of @id values. In this case the trick mode Adaptation Set is associated to all Adaptation Sets with the values of the @id.
- signal the playout capabilities with the attribute @maxPlayoutRate for each Representation in order to indicate the accelerated playout that is enabled by the signaled codec profile and level.
- If the Representation is encoded without any coding dependency on the elementary stream level, i.e. each sample is a SAP type 1, then it is recommended to set the @codingDependency attribute to FALSE.
- If multiple trick mode Adaptation Sets are present for one main Adaptation Set, then sufficient signaling should be provided to differentiate the different trick mode Adaptation Sets. Different Adaptation Sets for example may be provided as thumbnails (low spatial resolution), for fast forward or rewind (no coding dependency with @codingDependency set to false and/or lower frame rates), longer values for @duration to improve download frequencies or different @maxPlayoutRate values. Note also that the @bandwidth value should be carefully documented to support faster than real-time download of Segments.

4.10. Trick Mode for Live Services

4.10.1. Introduction

In certain use cases, along with the offering of the main content, a content author also wants to provide a trick mode version primarily of the video Adaptation Set along with the live content that can be used for rewind and fast forward in the time shift buffer of the Media Presentation. In section 3.2.9 signalling is introduced to flag and customize Adaptation Sets for Trick Modes. This clause provides additional service offering requirements and recommendations for trick modes in case of a live service. Typically, a reduced frame rate Representation or an I-frame only version is provided for supporting such trick mode operations.

4.3.2.2.5. Representation Information

Based on such an MPD at a specific time *NOW*, a list of Segments contained in a Representation in a Period *i* with Period start time *PSwc[i]* and Period end time *PEwc[i]* can be computed.

If the **SegmentTemplate.SegmentTimeline** is present and the **SegmentTemplate@duration** is not present, the **SegmentTimeline** element contains N_s **S** elements indexed with $s=1, \dots, N_s$, then let

- *ts* the value of the @timescale attribute
- *ato* is the sum of the values of the @availabilityTimeOffset attribute present on all levels that are processed in determining the URL for the corresponding segment, if present at all. If not present, it is zero.
- $t[s]$ be the value of @t of the s -th **S** element,
- $d[s]$ be the value of @d of the s -th **S** element
- $r[s]$ be,
 - if the @r value is greater than or equal to zero

Based on the above information, for each Representation r in a Period i , the segment availability start time $SAST[k,i,r]$, the segment availability end time of each segment $SAET[k,i,r]$, the segment duration of each segment $SD[k,i,r]$, and the URL of each of the segments, $URL[k,i,r]$ within one Period i be derived as follows using the URL Template function $URLTemplate(ReplacementString, Address)$ as documented in subsection 4.3.2.2.8:

<https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf>

4.3.2.2.1. Introduction

Based on an MPD including information as documented in Table 8 and available at time *NOW* on the server, a synchronized DASH client derives the information of the list of Segments for each Representation in each Period. This section only describes the information that is expressed by the values in the MPD. The generation of the information on the server and the usage of the information in the client is discussed in section 4.3.3 and 4.3.4, respectively.

MPD information is provided in subsection 4.3.2.2.3. The Period based information is documented in sub-section 4.3.2.2.4, and the Representation information is documented in sub-section 4.3.2.2.5.

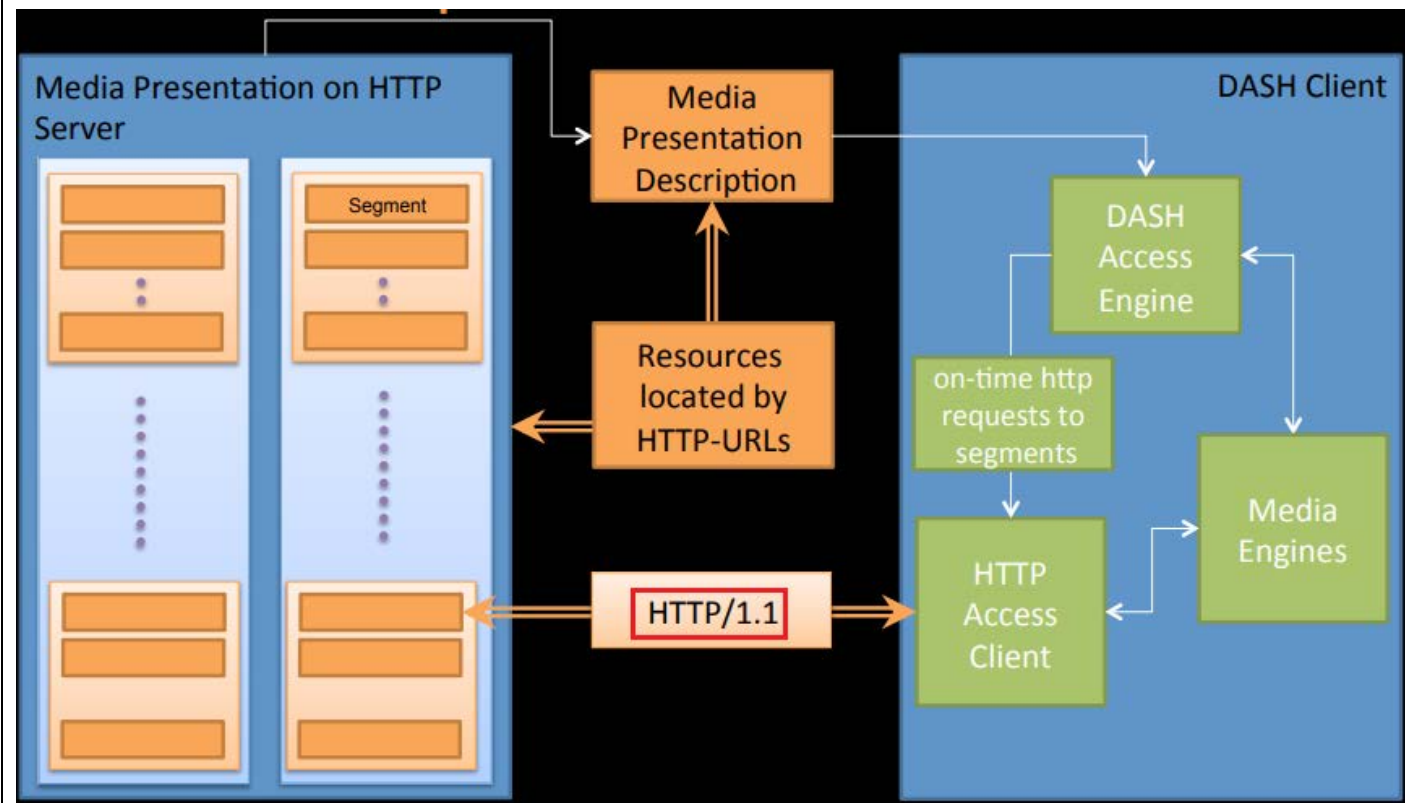
4.3.2.2.2. Definitions

The following definitions are relevant and aligned with ISO/IEC 23009-1:

- available Segment is a Segment that is accessible at its assigned HTTP-URL. This means that a request with an HTTP GET to the URL of the Segment results in a reply of the Segment and 2xx status code.
- valid Segment URL is an HTTP-URL that is promised to reference a Segment during its Segment availability period.
- *NOW* is a time that is expressing the time on the content server as wall-clock time. All information in the MPD related to wall-clock is expressed as a reference to the time *NOW*.

- as a single Segment, where Subsegments are aligned across Representations within an Adaptation Set. This permits scalable and efficient use of HTTP servers and simplifies seamless switching. This is mainly for on-demand use cases.
- as a sequence of Segments where each Segment is addressable by a template-generated URL. Content generated in this way is mainly suitable for dynamic and live services.

<https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf>



	<p>https://www.w3.org/2011/09/webtv/slides/W3C-Workshop.pdf</p> <p><u>In order to describe the temporal and structural relationships between segments, MPEG-DASH introduced the so-called Media Presentation Description (MPD).</u> The MPD is an XML file that represents the different qualities of the media content and the individual segments of each quality with HTTP Uniform Resource Locators (URLs). This structure provides the binding of the segments to the bitrate (resolution, etc.) among others (e.g., start time, duration of segments). <u>As a consequence, each client will first request the MPD that contains the temporal and structural information for the media content and based on that information it will request the individual segments that fit best for its requirements.</u></p> <p>https://bitmovin.com/dynamic-adaptive-streaming-http-mpeg-dash/</p>
transmission of the HTTP GET request to a source appliance; and	The Standard practices transmission of the HTTP GET request (e.g., the HTTP request by the DASH client device) to a source appliance (e.g., DASH server).

3.4.2. Server Requirements and Guidelines

HTTP Servers serving segments should support suitable responses to byte range requests (partial GETs).

If an MPD is offered that contains Representations conforming to the ISO BMFF On-Demand profile, then the HTTP servers offering these Representations shall support suitable responses to byte range requests (partial GETs).

HTTP Servers may also support the syntax using Annex E of 23009-1 using the syntax of the second example in Annex E.3,

BaseURL@byteRange="\$base\$?\$query\$&range=\$first\$-\$last\$"

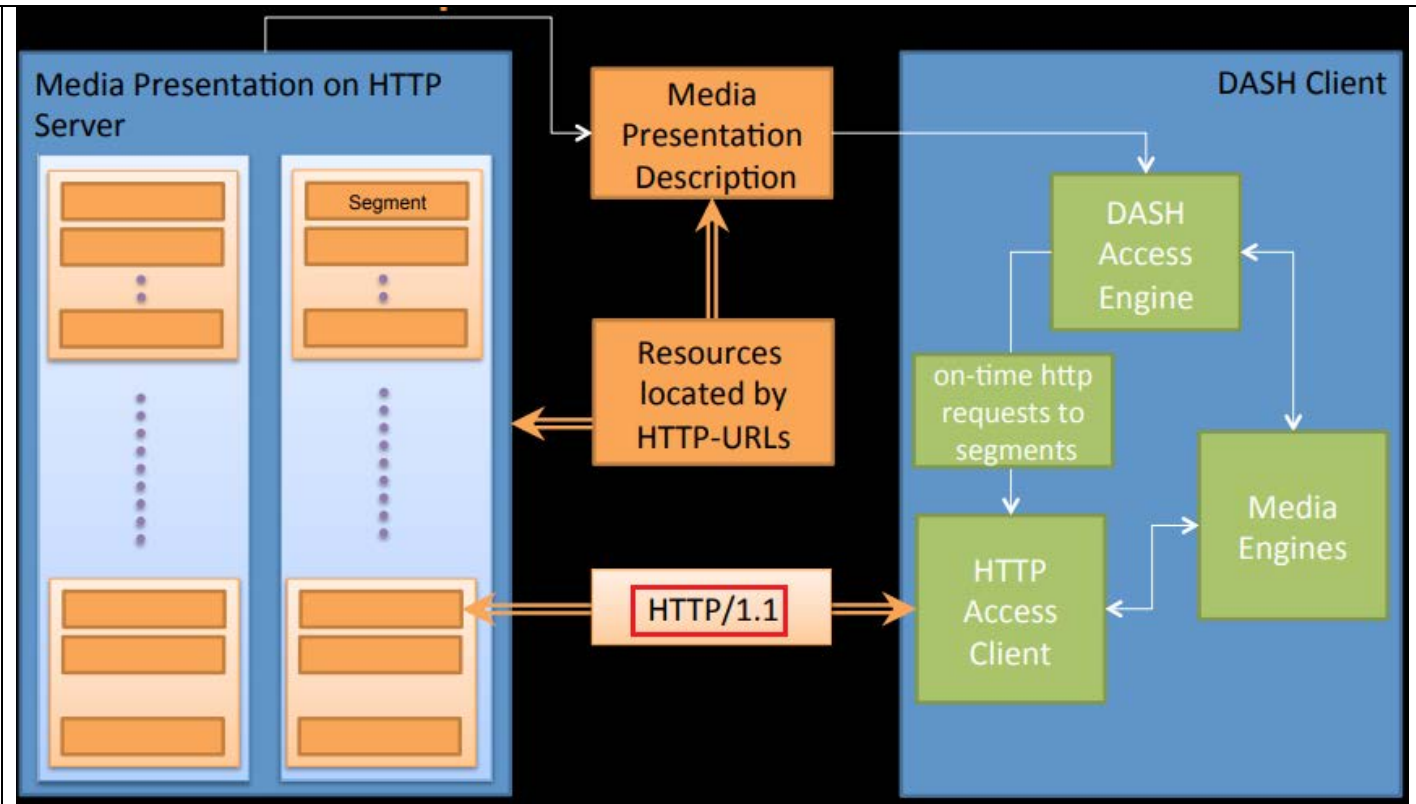
<https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf>

DASH Client

A DASH client retrieves and plays the video content using these steps: howstuffworks.com

1. The client downloads and reads the **Media Presentation Description (MPD)** to get important information, such as the content locations, segment encodings, resolution, minimum and maximum bandwidths, accessibility features like closed captioning and content restrictions (such as DRM).
2. The client selects an appropriate segment encoding and begins streaming the content through a series of HTTP requests. The server creates and encodes each segments on demand for each request, all from the same source.
3. The client buffers data as it's downloaded, while also keeping track of fluctuations in the connection bandwidth. If necessary, the client automatically changes to a different segment encoding (from those listed by the MPD) that's more compatible with the current bitrate. This ensures the client maintains a sufficient buffer throughout the video without downloading more data than you need.

https://www.bogotobogo.com/VideoStreaming/mpeg_dash.php



<https://www.w3.org/2011/09/webtv/slides/W3C-Workshop.pdf>

discontinuous transmission, in sections, of selected video frames of an original encoded video stream from the source appliance to a destination appliance in a HTTP response using an extended HTTP chunked transfer encoding mode, in which the selected encoded video frames for the fast search operation are transported in respective chunks, wherein each chunk includes one complete respective selected encoded video frame in a second part and information about a starting time, as located in the original encoded video stream, of the respective selected video frame in a first part, wherein the second part is different from the first part and the information about a starting time of the respective selected video frame being positioned in a commentary line of the first part.

The Standard practices discontinuous transmission (e.g., discontinues transmission of media segments during trick mode operation), in sections (e.g., media segments), of selected video frames (e.g., selected video frames in trick mode which skips other frames) of an original encoded video stream from the source appliance (e.g., HTTP server) to a destination appliance (e.g., DASH client) in an HTTP response using an extended HTTP chunked transfer encoding mode (e.g., transferring video data in segments or chunks in encoding format), in which the selected encoded video frames for the fast search operation (e.g., trick mode operation such as fast forward or rewind) are transported in respective chunks (e.g., segments or chunks), wherein each chunk includes one complete respective selected encoded video frame (e.g., a segment that comprises an I-frame) in a second part and information about a starting time (e.g., earliest presentation time EPT), as located in the original encoded video stream (Representation/Segments for trick mode are time-aligned with the main Adaption sets, e.g., the original video stream, which enables switching between the trick mode representation and the normal play rate representations), of the respective selected video frame (e.g., the selected video data frame in trick mode) in a first part, wherein the second part is different from the first part and the information about a starting time of the respective selected video frame being positioned in a commentary line of the first part (e.g., earliest_presentation_time field of the Segment Index).

4.10. Trick Mode for Live Services

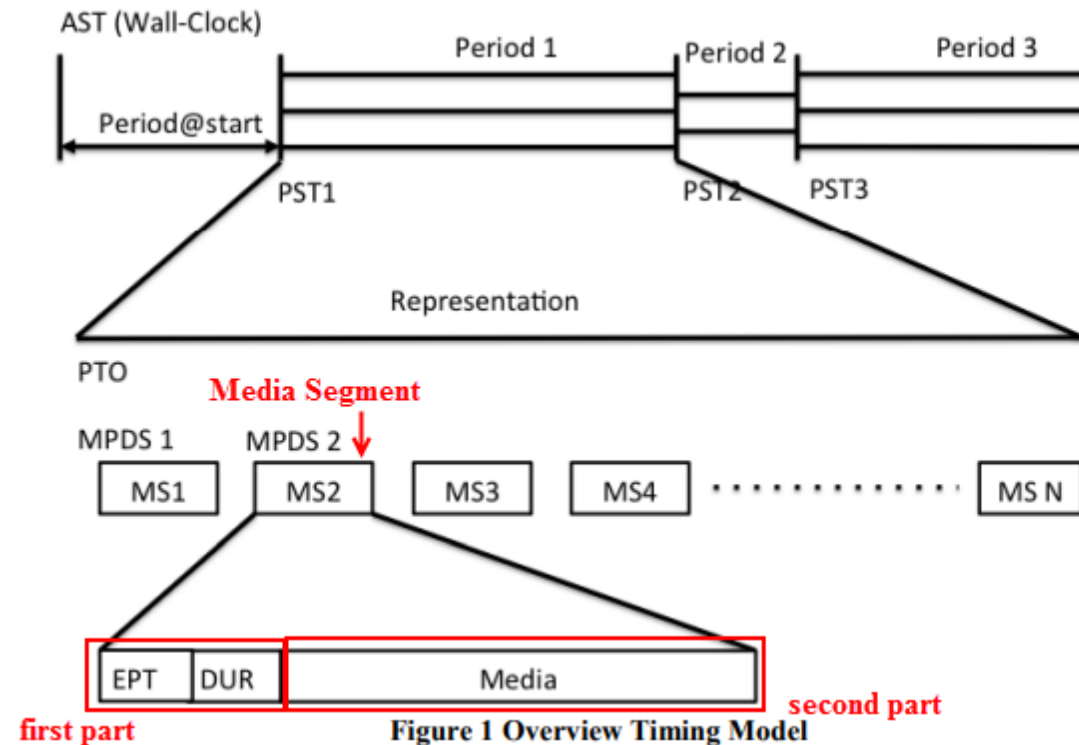
4.10.1. Introduction

In certain use cases, along with the offering of the main content, a content author also wants to provide a trick mode version primarily of the video Adaptation Set along with the live content that can be used for rewind and fast forward in the time shift buffer of the Media Presentation. In section 3.2.9 signalling is introduced to flag and customize Adaptation Sets for Trick Modes. This clause provides additional service offering requirements and recommendations for trick modes in case of a live service. Typically, a reduced frame rate Representation or an I-frame only version is provided for supporting such trick mode operations.

<https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf>

In On Demand Profile Periods, Sub-Representations in combination with Index Segments and Subsegment Index boxes may be used for efficient trick mode implementation. Sub-Representations in combination with Index Segments containing Segment Index boxes ('sidx') and Subsegment Index boxes ('ssix') may be used for trick mode implementation. Given a Sub-Representation with the desired @maxPlayoutRate, ranges corresponding to SubRepresentation@level all level values from SubRepresentation@dependencyLevel may be extracted via byte ranges constructed from the information in Subsegment Index Box. These ranges can be used to construct more compact HTTP GET request.

https://www.etsi.org/deliver/etsi_ts/103200_103299/103285/01.02.01_60/ts_103285v010201p.pdf



Within a Representation, each Segment is assigned an MPD start time and MPD duration according to ISO/IEC 23009-1 (more details for dynamic services are provided in section 4). These two values can be computed from the MPD and provide approximate times for each segment that are in particular useful for random access and seeking.

In addition, each segment has an internal sample-accurate presentation time. Therefore, each segment has a media internal earliest presentation time *EPT* and sample-accurate presentation duration *DUR*.

For each media segment in each Representation the MPD start time of the segment should approximately be $EPT - PTO$. Specifically, the MPD start time shall be in the range of $EPT - PTO - 0.5 * DUR$ and $EPT - PTO + 0.5 * DUR$ according to the requirement stated above.

Each Period is treated independently. Details on processing at Period boundaries are provided in ISO/IEC 23009-1. One example is, that for time code wrap-around a new Period is expected to be added that restarts at presentation time 0.

The value of **Period@start** for an ad can be chosen to coincide with an insertion point in the live stream by setting **Period@start** to a presentation time duration equal to the UTC time difference between **@availabilityStartTime** and the scheduled encoding time of the insertion point in the live stream.

The earliest presentation time may be estimated from the MPD using the segment availability start time minus the segment duration announced in the MPD.

The earliest presentation time may be accurately determined from the Segment itself.

If the Segment Index is present then this time is provided in the earliest presentation_time field of the Segment Index. To determine the presentation time in the Period, the value of the attribute @presentationTimeOffset needs to be deducted.

4.5.3.3. Extended Segment Information

The DASH client shall download the selected Representation and shall parse the segment at least up to the first 'moof' box. The DASH client shall parse the segment information and extract the following values:

- ept the earliest presentation time of the media segment
 - if the Segment Index is present use the Segments Index
 - if not use the baseMediaDecodeTime in 'tfdt' of the first movie fragment as the earliest presentation time
- dur the media presentation duration of the media segment
 - if the Segment Index is present use the Segments Index
 - if not use aggregated sample durations of the first movie fragment as the duration
- annotate each Adaptation Set with an **EssentialProperty** descriptor or **SupplementalProperty** descriptor with URL "http://dashif.org/guidelines/trickmode" and the @value the value of @id attribute of the Adaptation Set to which these trick mode Representations belong. The trick mode Representations must be time-aligned with the Representations in the main Adaptation Set. The value may also be a white-space separated list of @id values. In this case the trick mode Adaptation Set is associated to all Adaptation Sets with the values of the @id.

In both cases (Sub)Segments begin with Stream Access Points (SAPs) of type 1 or 2 [7], i.e. regular IDR frames in case of video. In addition, (Sub)Segments are constrained so that for switching video Representations within one Adaptation Set the boundaries are aligned without gaps or overlaps in the media data. Furthermore, switching is possible by a DASH client that downloads, decodes and presents the media stream of the come-from Representation and then switches to the go-to Representation by downloading, decoding and presenting the new media stream. No overlap in downloading, decoding and presentation is required for seamless switching of Representations in one Adaptation Set.

Content in one Adaptation Set is constrained to enable and simplify switching across different Representations of the same source content. General Adaptation Set constraints allow sequencing of Media Segments from different Representations (“bitrate switching”) prior to a single audio or video decoder, typically requiring the video decoder to be reset to new decoding parameters at the switch point, such as a different encoded resolution or codec profile and level.

Bitstream Switching Adaptation Set constraints allow a switched sequence of Media Segments to be decoded without resetting the decoder at switch points because the resulting Segment stream is a valid track of the source type, so the decoder is not even aware of the switch. In order to signal that the Representations in an Adaptation Set are offered under these constraints, the attribute **Ad-**

<https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf>